2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI) | 979-8-3503-5118-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/IRI62200.2024.00069

Multi-Eyes: A Framework for Multi-User Eye-Tracking using Webcameras

Bhanuka Mahanama, Vikas Ashok, Sampath Jayarathna
Department of Computer Science, *Old Dominion University*, Norfolk, VA, USA
bhanuka@cs.odu.edu, vganjigu@cs.odu.edu, sampath@cs.odu.edu

***Abstract*—The human gaze provides informative cues on human behavior during interactions in multi-user environments. However, capturing this gaze information using traditional eye trackers often requires complex and costly experimental setups. Furthermore, conventional eye-tracking algorithms are catered for single-user scenarios and cannot be used for multi-user environments. We propose Multi-Eyes, a commodity webcam-based solution offering scalability and cost-efficiency while leveraging the advancements in deep learning for capturing multi-user gaze. Multi-Eyes propose a three-step multi-user eye tracking framework that (1) detects gaze subjects, (2) estimates gaze, and (3) maps gaze-to-screen with a scalable, memory, and parameter-efficient disentangled gaze estimation model. We evaluate the gaze estimation model using two publicly available datasets and the framework's utility through a joint-attention case study. Our proposed architecture achieves the lowest gaze error of 4.33, while the case study demonstrates the feasibility of the Multi-Eyes for multi-user interactions and joint attention with comparable results to the state-of-the-art.**

***Index Terms*—Eye Tracking, Multi-user, Deep Learning, Joint Attention**

I. Introduction

The gaze provides insight into human behavior ranging from human-computer interaction [1], behavioral sciences [2], and various other domains [3]. User interactions often happen in collaborative environments though many studies in eye tracking fail to capture collaborative behaviors primarily due to studies being conducted in isolation. This can be attributed to a couple of factors. First, even though eye trackers can accurately capture the gaze of a single user, they cannot capture the gaze information of more than one participant simultaneously. Second, conventional eye-tracking algorithms that leverage a single user for multi-user eye-tracking fail to scale due to the requirement of a dedicated device per participant, compounding the complexity and cost of the experimental setup.

Eye tracking using commodity hardware (i.e., web camera) provides a cost-efficient alternative to the limitations posed by conventional eye trackers. Recent advancements in computer vision have been steering a plethora of recent developments in appearance-based gaze estimations [4], [5], [6]. Combined with large-scale datasets[5], [7], these allow models with improved feature extraction and accuracy. Despite being conceptually and technologically promising, these approaches depend on the computational capacity of the platform[8]. Therefore consistent performance requires scalable models balancing the complexity and capacity.

While commodity hardware may offer a cost-effective option for multi-user eye-tracking, scalable and efficient models for appearance-based eye-tracking still need to be developed. This particularly plays a vital role in multi-user environments where computation demand grows proportional to the number of users. Our study investigates how to leverage recent advancements to develop **a low-cost appearance-based multi-user eye-tracking system using deep learning techniques**. Our contributions are three-fold;

1) We introduce a family of scalable gaze models;
2) We use these models to design a multi-user eye-tracking methodology using low-cost commodity hardware;
3) We demonstrate the feasibility and utility of our approach using a case study on joint attention and evaluate experimental results.

II. Related Work

In this section, we review related literature on gaze estimation techniques followed by multi-user eye tracking.

A. Gaze Estimation

Gaze estimation methodologies are broadly classified as model-based or appearance-based methods[9], [6], [10]. Model-based approaches use landmarks to find ocular or facial features and employ a geometric model of the eye [11], [12] or face [13], [14] to estimate the gaze direction. These methods rely heavily on correctly identifying landmarks such as pupil center [11], [15]. For this purpose, these methods use other[11] or incorporate additional [10] modalities, such as infrared lighting [10].

In contrast, appearance-based approaches utilize images to estimate the gaze directions using either ocular[6], [16] or facial images[17], [4], [18], forming a mapping function between the image and the gaze directions [9]. This eliminates the requirement of intermediate computation of facial landmarks. Based on the technique employed, these approaches can be further classified into conventional or deep learning approaches [10]. Conventional appearance-based approaches utilize image processing techniques (e.g., histogram equalization[19]) combined with machine learning models (e.g., support vector machines[20], linear regression [16], [21], or neural networks [19]) to estimate the gaze. Despite the simplicity of the approach, these models are often constrained by the capacity of the feature extractor and the complexity of the gaze estimation model.

979-8-3503-5118-7/24/$31.00 ©2024 IEEE
DOI 10.1109/IRI62200.2024.00069

Instead of relying on generic features or dimensionality reduction techniques, deep learning methods approach this problem by detecting features and mapping them to the gaze estimation[10]. Recent studies in deep learning gaze estimation have shown Convolutional Neural Networks (CNNs) to be an excellent candidate for appearance-based gaze estimation[22], [23]. In order to achieve improved accuracy, CNNs typically scale up by adding more layers [24], which can often lead to bulkier, deeper CNN models. Despite the performance gain, these models tend to be computationally expensive due to their complexity.

Despite the popularity of deep learning-based models in general computer vision applications, the wide adoption of mobile devices has led to the development of computationally efficient CNNs. Mobile-oriented CNN models such as MobileNet[25] and ShuffleNet[26] attempt to address the issue through computationally efficient layers. However, the scaling of these models in resource-rich environments remains arbitrary and often limited to one of the three dimensions: resolution, depth, and width. As a result, despite the efficiency of resource-constrained environments, they fail to exploit the benefits of resource-rich environments due to a lack of systematic scaling. In contrast, EfficientNets[27] are a class of CNNs built around the principles of systematic scaling. As a result, an application developed utilizing EfficientNets can scale per the device's capabilities.

B. Multi-user Eye-tracking

Despite the wide adoption of eye tracking for single-user experiments [28], the concept of multi-user eye tracking remains relatively less explored across existing domains. Studies that use multi-user eye tracking are of two main types: time-sharing and space-sharing [10]. The time-sharing approaches [29] combine the gaze information of multiple users spanning non-overlapping time windows. In comparison, space-sharing approaches [30], [31], [32] estimate the gazes of multiple users concurrently [10].

The first challenge for the space-sharing approach is the lack of specialized hardware for the purpose. Even though eye trackers excel in estimating gaze for single-user studies, they cannot be directly used for space-sharing setups as they cannot track more than one person. A straightforward approach to overcome the issue is to use a dedicated device per participant [33], [31]. Despite the simplicity, the solution can lead to multiple issues. There can be interferences among the eye trackers, leading to incorrect data, which can be mediated by imposing strict restrictions on the movement of the users in the setup. However, the setup will not scale well for large-scale experiments, driving the cost of the experimental setup.

III. Methodology

A. Gaze Model

Intuitively we can identify a face image patch to comprise two feature forms: gaze-defining, such as ocular region features, and non-gaze-defining features, such as skin complexion. We extend the idea to low-dimensional representations of the image, which we model using standard autoencoders. A standard autoencoder comprises an encoder that transforms the data into a low-dimensional latent representation and a decoder that uses the representation to reconstruct the input. During training, an autoencoder learns an entangled representation without modification, meaning we cannot classify each dimension between two feature types. To overcome the entanglement, we introduce an architectural modification to the auto-encoders and form the disentangled gaze models.

Our model architecture (see Figure 1) uses an encoder that transforms a given image into an encoded representation as $E : x \rightarrow e$ and a decoder $D : e \rightarrow \tilde{x}$ that reconstructs an approximation of the original input such that $x \approx \tilde{x} = D(E(x))$. In order to disentangle the representation, we consider the latent space to comprise two feature forms, gaze-defining encodings (e_g) and non-gaze-defining encodings (e_f) such that $E(x) = \{e_f(x); e_g(x)\}$. For enforcing the disentanglement to the model, we introduce an additional decoder - Gaze decoder $G : e_g \rightarrow g$, which decodes gaze-defining encodings into the target gaze descriptor (e.g., gaze angles, gaze positions, or gaze categories). For a latent representation generated by the encoder to be of $N \in \mathbb{N}$ elements, we define a hyper-parameter $\kappa \in [0, N]$- the number of elements allocated for the gaze-dependent features in the latent space, termed as disentanglement of our model. Therefore, the dimensionalities of e_g and e_f becomes κ and $N - \kappa$ respectively.

Our architectural modification provides two additional benefits in addition to enforcing disentanglement. First, image reconstruction acts as a form of regularization to the gaze estimation network (E, G), thus preventing overfitting. Second, because we use only part of the latent space for estimating gaze, we get a comparatively lesser number of parameters for gaze estimation compared to models that utilize the entire latent space.

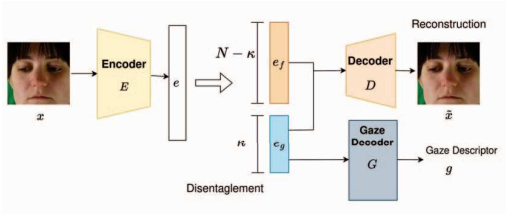

Fig. 1: Proposed disentangled model architecture for gaze estimation.

Our overall loss for the model comprises two loss terms considering the task rendered by the model. First, considering the image reconstruction of the decoder D, we define the reconstruction loss L_r as,

$$L_r(x, \tilde{x}) = \frac{1}{|x|} \sum |u_i - \tilde{u_i}| \quad (1)$$

where u_i and $\tilde{u_i}$ are corresponding pixels from two images.

Next, we define the Gaze error (L_g), considering the gaze direction in the image and the gaze estimation from the gaze decoder G by,

$$L_g(g, \tilde{g}) = \frac{1}{|g|} \sum |v_i - \tilde{v_i}| \quad (2)$$

where g is the ground truth gaze descriptor, $\tilde{g}$ estimated gaze descriptor, and v_i, and $\tilde{v_i}$ are elements of the gaze descriptors. Finally, we combine the error terms using two hyper-parameters $L = \lambda_r L_r + \lambda_g L_g$, defining weights for each type of loss in the model.

Considering different combinations of the hyper-parameters of the model λ_r, λ_g, and κ, we can build a family of models that are both disentangled and aware of gaze features leveraging the disentanglement. For instance, we can arrive at a classical auto-encoder type model with $\kappa = 0$ and $\lambda_g = 0$ ($\lambda_r > 0$). On the other hand, we derive a naive gaze estimation model with $\kappa = N$ and $\lambda_r = 0$ ($\lambda_g > 0$).

We use the publicly available ETH X-Gaze dataset [17] of over one million high-resolution images of varying gaze under extreme head poses to train the model. The dataset consists of facial images of 110 participants, collected using a custom hardware setup with 18 digital SLR cameras, an adjustable illumination setup, and a calibrated system to record ground-truth gaze targets. We use an EfficientNet architecture-based CNN that takes input images of shape (224×224) and produces a latent representation of $(N \times 1)$ as the encoder (E) in our model with empirically chosen $N = 64$ for our experiments. We use a deconvolutional neural network that uses the latent representation and reconstructs the facial image as the decoder (D). The gaze decoder (G) comprises a fully connected neural network that estimates the gaze in the form of pitch and yaw angles. We use Adam optimizer [34] with a linearly decaying learning rate starting from 0.001, decays to 0.0001 throughout 50 epochs, and 80-20 training and validation split in the study during the training process. When forming the validation splits, we use the participants as the selection criteria for the validation split, ensuring the images used in validation remain unknown to the model.

B. Multi-user Eye-Tracking

Our proposed multi-user eye-tracking architecture (see Figure 2) employs a three-step process, (1) gaze subject detection, (2) gaze estimation, and (3) gaze-to-screen mapping to estimate the gaze position (i.e., gaze coordinates in the display) of the participants.

Gaze Subject Detection: For the simplicity of our prototype, we divide the camera images into regions, referred to as user-designated regions, where we expect the user to be present during the experiment. Further, we assume no occlusions exist between the camera and the user and only one user to present in each region. We expect to eliminate redundant operations such as face detection or occlusion detection. We split the image vertically into two regions where we expect the two participants to be present. Then, we utilize the Facemesh [35] model to detect the faces in the image region and establish the bounding box using the centroid of the detected landmarks. Finally, we crop and generate the face patches for the gaze estimation step.

Gaze Estimation: Here, we process the images using a model variant to estimate the gaze directions expressed as pitch and yaw angles with respect to the detected face. Since the image patches can be of different sizes depending on the user's distance from the camera, we use bilinear interpolation [36] to resize each facial image to match the specifications of the estimation model. We select and use a model from the model variants discussed earlier based on gaze estimation and inferencing throughput. To orient model performance on gaze estimation, we empirically select $N = 64$, $\kappa = N$, and train the model with $\lambda_g = 1$ and $\lambda_r = 1$. Considering the real-time inferencing performance, we use the model comprising EfficientNet-B0 [27] as the encoder.

Gaze-to-Screen Mapping: Our approach uses the encoded face position to model the relationship between gaze directions and on-screen positions. Here, we propose a grid-like encoding scheme to represent the position derived using the pinhole camera model, assuming the camera remains stationary relative to the interaction surface.

For our calculations, we consider the face of a person at (x_p, y_p, z_p) with dimensions $(\Delta x_p, \Delta y_p, \Delta z_p)$ in the world coordinate frame, projecting an image of size $(\Delta u, \Delta v)$ at (u, v) on camera coordinate system. We derive the relationships $u = \frac{f}{z_p} x_p$ and $\Delta u = \frac{\Delta x_p}{z_p} f$ using the pinhole camera model, where f represents the focal distance.

Similarly, we derive a similar relationship for v and Δv, indicating the possibility of encoding the facial location in 3D space using the projection on the image. For this purpose, we first form a mask M of the input image size with i, and jth value defined as,

$$M_{i,j} = \begin{cases} 1 & \text{if } u \leq i \leq u + \Delta u \text{ and } v \leq j \leq v + \Delta v \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Then we perform average pooling on the mask with the same pool size and stride to derive the derived mask to obtain the position encoding of size $n \times n$.

Finally, we combine the gaze direction estimate (g) and the encoded face positions (P) to estimate the gaze locations on the screen $(s = (s_x, s_y))$ using a mapping function $S : (P, g) \rightarrow s$. For modeling the mapping function, we assume a nonlinear relationship between the variables modeled through a multi-layer neural network trained during the calibration phase of the application. Even though we can change the sensitivity by choosing different values for n, we use $n = 2$ in our prototype setup for simplicity. Moreover, in each session, the proctor monitors the training and validation errors during each calibration round to prevent overfitting.

IV. Results

A. Gaze Estimation

We first evaluate the model performance by implementing the encoder (E) by CNNs with EfficientNet architectures, a class of CNNs built for systematic scaling. We test and report the performance against the publicly available ETH XGaze

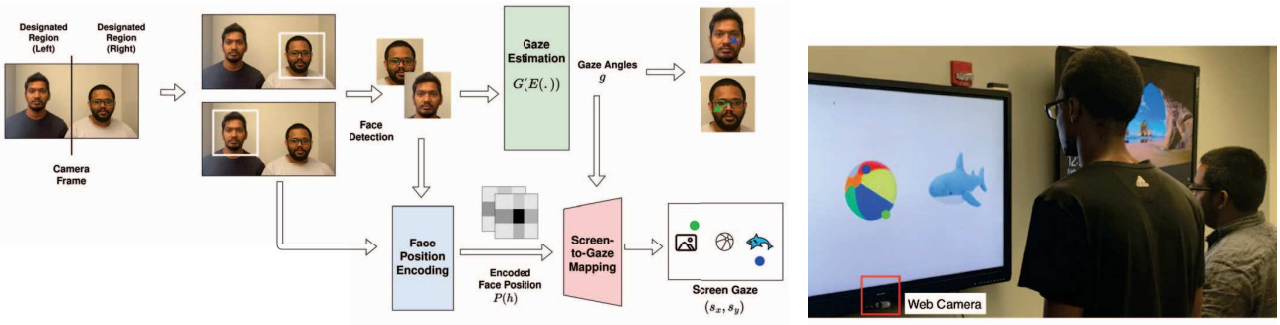

Fig. 2: Left: Proposed Multi-Eyes framework, Right: Joint attention experiment with two participants

[17] dataset across all model variants. We use $\kappa = N$ to allocate the entire latent space for gaze estimation and report the performance using the Mean Absolute Error (MAE) of the gaze angle estimation (see Table I). The results suggest that the series of models we used in the experiments provide improved accuracy by increasing the number of parameters. This allows us to identify efficient model sizes depending on the accuracy and the hardware capabilities required for a scalable multi-user eye-tracking system. However, it is essential to note that through each model configuration, we can derive more optimized models by adjusting hyper-parameters often leading to better-generalized models while achieving better parameter efficiency.

TABLE I: Evaluation of the Gaze Estimation Network using ETH X-Gaze Dataset[17] ($N = 64$, $\kappa = 64$ and $\lambda_r = 0$).

Model	Gaze Error	# Parameters
X-Gaze Baseline [17]	4.50	26M
EfficientNet-B0	5.22	4.3M
EfficientNet-B1	5.12	6.9M
EfficientNet-B2	5.05	9.1M
EfficientNet-B3	4.99	11M
EfficientNet-B4	4.85	18M
EfficientNet-B5	4.64	29M
EfficientNet-B6	4.64	42M
EfficientNet-B7	**4.34**	65M

We use the EfficientNet-B0 encoder-based model and explore the effect of hyperparameters λ_r and λ_g of the model. Here we use the $N = 64$ and $\kappa \in \{16, 32, 64\}$ and change the hyperparameters and report the results (see Table II). Here we explore the possibility of using the reconstruction as a form of regularization to the network and its effect on gaze estimation accuracy. The decrease in error for increments of the hyperparameter λ indicates where $\kappa \in \{16, 64\}$ that the model achieves more generalizability through regularization. In contrast, the experiments with $\kappa = 32$ show an opposite pattern with the decrease in λ corresponding to an increase in the accuracy of gaze estimation, with $\lambda_r = 0.01, \kappa = 64$ yielding the least gaze estimation error.

Considering the utility of the proposed gaze model across hardware with varying computational capabilities requires different optimizations to leverage the hardware capabilities of the host platform. In our experiments, we explore the effect of due to quantization, where we execute the model with lower

TABLE II: Effect of Hyper Parameters on the estimation. EfficientNet-B0 model ($N = 64$, $\lambda_g = 1$)

	Regularization (λ_r)		
	0.01	0.1	1
$\kappa = 16$	5.3520	5.2854	5.1936
$\kappa = 32$	5.1340	5.1815	5.4166
$\kappa = 64$	5.1604	5.1725	**5.1012**

precision parameters by discretizing the model parameters. Quantization allows to compress the model and run with lower computations. Our study considers float16 quantization, transforming the mode parameters from float64 to float16. We conduct the study in two steps; first, we use the same models used to study the effect of hyperparameters and analyze the effect of quantization. Next, we compare the performance against models trained with emulated quantization in the forward pass (quantization-aware).

Our results (see Table III) indicate that the quantization of pre-trained models yields mixed results between different combinations of model parameters, with the highest achieved by $\lambda_r = 0.01, \kappa = 32$. In comparison, the quantization-aware models (see Table IV) lead to higher gaze errors in similar model configurations, indicating that the additional step of emulating quantization did not improve estimation accuracy.

TABLE III: Effect of quantization on Gaze Error (+improvement/ -decline%) using EfficientNet-B0 ($N = 64$, $\lambda_g = 1$)

	Regularization (λ_r)		
	0.01	0.1	1
$\kappa = 16$	5.3504 (+0.45)	5.2854 (-0.11)	5.1926 (+0.01)
$\kappa = 32$	**5.1347** (-0.01)	5.1821 (-0.01)	5.4166 (+0.03)
$\kappa = 64$	5.1582 (+0.04)	5.1735 (-0.02)	5.4147 (**-6.15**)

TABLE IV: Effect of quantization on Mean Gaze Error using EfficientNet-B0 with emulated quantization on forward pass. ($N = 64$, $\lambda_g = 1$)

	Regularization (λ_r)		
	0.01	0.1	1
$\kappa = 32$	5.555	5.594	5.639
$\kappa = 64$	5.686	5.736	5.979

Next, we evaluate the knowledge transferability of the model against the publicly available Columbia-Gaze dataset [20] of 5,880 images of 56 people over varying gaze directions

and head poses. We pass each image in the dataset through Facemesh[35] to identify faces and evaluate the gaze estimation for each detected face. We observed no clear patterns between the model configuration and the gaze estimation error (see Table V). However, the results provide an estimate of the model's generalizability for potential application in real-world studies, which can be improved through model scaling or calibration.

TABLE V: Cross-dataset evaluation of gaze estimation network using Columbia-Gaze Dataset [20] ($N = 64$, $\lambda_g = 1$)

	Regularization (λ_r)		
	0.01	0.1	1
$\kappa = 16$	6.091	5.548	7.177
$\kappa = 32$	6.211	6.021	**5.305**
$\kappa = 64$	6.027	7.107	6.229

B. Feasibility Study: Joint Attention Tracking

To demonstrate the feasibility of our framework, we designed an experimental setup (see Figure 2) using a 72" wall-mounted display and Logitech C270 web camera (720p/30fps, 55° dFoV) to capture multi-user gaze movements. We recruited five participants aged 20 - 27 (2 F, 3M) for the user study (3 pairs, 7 trials, within-subject) with normal or corrected-to-normal vision. We instructed participants to stand approximately 6 feet in front of the screen and adjusted the web camera based on their heights. We started our experiment by calibrating the eye tracker for each user separately. Here, the application selectively prompted each user to look at five given targets (top left, top right, bottom left, bottom right, and center of the screen) sequentially. At the prompt, the application collected gaze direction and encoded face location after a predefined delay to settle the users' gaze. The application collected 100 samples for each target before proceeding to the next target in the sequence. Before the experiment, we allowed the participants to test their calibration by presenting a test screen and asking them to fixate on selected positions.

Following the work of [37], each joint interaction trial consisted of a 1-minute five-point calibration step and a 1.5-2.5 minute joint attention task among two participants in front of the screen and one proctor across from the participants. The web camera was placed on the bottom center of the screen. Afterward, the proctor instructed the participants to look at one of the objects (photo frame, ball, or shark) displayed on the screen for 10 seconds. For each trial, the participants were given an object name to look at. The order of the objects was randomized among different pairs of participants.

Given the gaze position (i.e., x and y screen coordinators) of participants along with timestamps, we derived eye movement metrics indicative of joint attention, *fixation duration, fixation count,* and *time to first fixation* [38]. We derived eye movement are-of-interest (AOI) given three object boundaries and extracted feature sets within these AOIs.

In order to detect joint attention, we measured the time to the first fixation upon receiving an instruction. We also measured the number of fixations a subject made on the object upon receiving instruction and the fixation duration of subjects upon fixating on an object. Table VI presents the calculated eye-tracking metrics. Our results using commodity hardware are comparable to the joint attention tasks from prior work given conventional eye tracking configurations [38], [39], [40], [41].

TABLE VI: Eye Movement Measurements During the Joint Attention Tasks

Metric	Mean	Median	Std.
Time to First Fixation (seconds)	1.25	1.18	0.93
Fixation Count	7.26	5.50	6.72
Fixation Duration (seconds)	5.45	6.06	3.00

Further, we evaluated the *throughput* of our experimental setup considering the average frame rate across the joint attention experiments, using the number of gaze samples generated by each thread dedicated for each user. Our results indicate that without any optimizations or hardware acceleration, our setup achieved an average throughput of 17 frames per second (see Table VII) with a max throughput rate of 20 frames per second.

TABLE VII: Multi-eyes pipeline throughput during joint attention experiment

	Left Thread			Right Thread		
Session	Mean	SD	Max	Mean	SD	Max
Session 1	17.05	2.02	19	17.06	1.98	20
Session 2	16.84	2.18	19	17.18	1.76	21
Session 3	17.18	1.56	20	16.88	2.55	21

V. Conclusion

In this paper, we present Multi-Eyes, a framework for performing multi-user gaze estimation using commodity hardware. The proposed approach's lightweight EfficientNet model allows gaze estimation to be deployed to a wide range of devices.

Our results show the potential for the off-the-shelf hardware resources to perform gaze estimation during multi-user interactions. Our prototype did not use hardware acceleration such as GPUs, CPU-based optimizations, or software-based optimizations, which are possible avenues to improve our application's throughput. However, the prototype in its current form would be sufficient for applications that require approximate gaze positions with low sampling rates. In the future, we expect to incorporate application-level optimizations to further improve scalability during multi-user interactions, and comprehensive end-to-end evaluation of our pipeline.

Acknowledgment

This work was supported in part by NSF 2045523.

References

[1] A. Sharma and P. Abrol, "Eye gaze techniques for human computer interaction: A research survey," *International Journal of Computer Applications*, vol. 71, no. 9, 2013.

[2] S. V. Shepherd, "Following gaze: gaze-following behavior as a window into social cognition," *Frontiers in integrative neuroscience*, vol. 4, p. 5, 2010.

[3] B. Mahanama, Y. Jayawardana, S. Rengarajan, G. Jayawardena, L. Chukoskie, J. Snider, and S. Jayarathna, "Eye movement and pupil measures: A review," *Frontiers in Computer Science*, vol. 3, p. 733531, 2022.

[4] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2176–2184.

[5] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 162–175, 2017.

[6] B. Mahanama, Y. Jayawardana, and S. Jayarathna, "Gaze-net: appearance-based gaze estimation using capsule networks," in *Proceedings of the 11th Augmented Human International Conference*, 2020, pp. 1–4.

[7] P. Kellnhofer, A. Recasens, S. Stent, W. Matusik, and A. Torralba, "Gaze360: Physically unconstrained gaze estimation in the wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6912–6921.

[8] B. Mahanama, "Multi-user eye-tracking," in *2022 Symposium on Eye Tracking Research and Applications*, 2022, pp. 1–3.

[9] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 478–500, 2009.

[10] P. Pathirana, S. Senarath, D. Meedeniya, and S. Jayarathna, "Eye gaze estimation: A survey on deep learning-based approaches," *Expert Systems with Applications*, vol. 199, p. 116894, 2022.

[11] M. Kassner, W. Patera, and A. Bulling, "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*, 2014, pp. 1151–1160.

[12] O. Palinko, F. Rea, G. Sandini, and A. Sciutti, "Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 5048–5054.

[13] L. Fridman, P. Langhans, J. Lee, and B. Reimer, "Driver gaze region estimation without use of eye movement," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 49–56, 2016.

[14] M. X. Huang, T. C. Kwok, G. Ngai, H. V. Leong, and S. C. Chan, "Building a self-learning eye gaze model from user interaction data," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 1017–1020.

[15] E. Wood and A. Bulling, "Eyetab: Model-based gaze estimation on unmodified tablet computers," in *Proceedings of the symposium on eye tracking research and applications*, 2014, pp. 207–210.

[16] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "Webgazer: Scalable webcam eye tracking using user interactions," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI, 2016, pp. 3839–3845.

[17] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, and O. Hilliges, "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.

[18] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "It's written all over your face: Full-face appearance-based gaze estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 51–60.

[19] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, 2010, pp. 3739–3744.

[20] B. Smith, Q. Yin, S. Feiner, and S. Nayar, "Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction," in *ACM Symposium on User Interface Software and Technology (UIST)*, Oct 2013, pp. 271–280.

[21] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 10, pp. 2033–2046, 2014.

[22] V. Nagpure and K. Okuma, "Searching efficient neural architecture with multi-resolution fusion transformer for appearance-based gaze estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 890–899.

[23] X. Guo, Y. Wu, J. Miao, and Y. Chen, "Litegaze: Neural architecture search for efficient gaze estimation," *Plos one*, vol. 18, no. 5, p. e0284814, 2023.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[26] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition* 2018, pp. 6848–6856.

[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[28] B. Mahanama, Y. Jayawardana, S. Rengarajan, G. Jayawardena, L. Chukoskie, J. Snider, and S. Jayarathna, "Eye movement and pupil measures: A review," *frontiers in Computer Science*, vol. 3, p. 733531, 2022.

[29] Y. Sugano, X. Zhang, and A. Bulling, "Aggregaze: Collective estimation of audience attention on public displays," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 821–831.

[30] F. Broz, H. Lehmann, C. L. Nehaniv, and K. Dautenhahn, "Mutual gaze, personality, and familiarity: Dual eye-tracking during conversation," in *2012 IEEE RO-MAN: The 21st IEEE international symposium on robot and human interactive communication*. IEEE, 2012, pp. 858–864.

[31] B. Mahanama, M. Sunkara, V. Ashok, and S. Jayarathna, "Disetrac: Distributed eye-tracking for online collaboration," in *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval* 2023, pp. 427–431.

[32] Y. Abeysinghe, B. Mahanama, G. Jayawardena, M. Sunkara, V. Ashok, and S. Jayarathna, "Gaze analytics dashboard for distributed eye tracking," in *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2023, pp. 140–145.

[33] S. Cheng, J. Wang, X. Shen, Y. Chen, and A. Dey, "Collaborative eye tracking based code review through real-time shared gaze visualization," *Frontiers of Computer Science*, vol. 16, no. 3, pp. 1–11, 2022.

[34] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[35] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile gpus," *arXiv preprint arXiv:1907.06724*, 2019.

[36] E. J. Kirkland, "Bilinear interpolation," in *Advanced Computing in Electron Microscopy*. Springer, 2010, pp. 261–263.

[37] S. Guo, E. Ho, Y. Zheng, Q. Chen, V. Meng, J. Cao, S. Wu, L. Chukoskie, and P. Cosman, "Using face and object detection to quantify looks during social interactions," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 1144–1148.

[38] D. Ambrose, D. E. MacKenzie, P. Ghanouni, and H. F. Neyedli, "Investigating joint attention in a guided interaction between a child with asd and therapists: A pilot eye-tracking study," *British Journal of Occupational Therapy*, p. 0308022620963727, 2020.

[39] R. J. K. Jacob and K. S. Karn, "Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises," in *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, J. Hyönä, R. Radach, and H. Deubel, Eds. Amsterdam, The Netherlands: Elsevier Science, 2003, pp. 573–605.

[40] P. M. Fitts, R. E. Jones, and J. L. Milton, "Eye Movements of Aircraft Pilots During Instrument-Landing Approaches," *Aeronautical Engineering Review*, vol. 9, no. 2, pp. 24–29, 1950.

[41] M. A. Just and P. A. Carpenter, "Eye Fixations and Cognitive Processes," *Cognitive Psychology*, vol. 8, no. 4, pp. 441–480, October 1976.